

PROGRAMMABLE KEYBOARD STATUS A meeting of the FCC is scheduled around Sept.15, at which time the TI petition will be discussed. Bally currently feels that regardless of the decision, they will not be able to have a keyboard in production by the end of the year.

SURVEY The subscriber survey has resulted in a fair turnout of responses from those interested in a keyboard/memory addition, and a number of useful suggestions as well. What is evolving now is a unit that could have 16K of RAM that would accept a cassette input of the operating program, be it BASIC, COBOL, FORTRAN, or whatever(your choice) which would be loaded in about 2 minutes at 1200 baud (while the picture tube was warming up), plus an additional 8 or 16 K of onboard R M for the user's programs. There would be space to add more R M by chip insertion(especially the 8K version), plus connectors to allow outside memory addition. Serial and parallel ports would be available for the addition of other peripherals. Those who responded to the survey will be kept up to date.

BLACK BOX GAME enclosed is a sort of Battleship game where the computer hides some "atoms" in a grid and you have to locate them. Use the diagram for clues.

TUTORIAL ON SOUND adds more material from Chuck Thomka to last issue's discourse

SLOT MACHINE CORRECTION requires the addition of a comma to the very end of line 1515 to stop the scrolling.

HIG LETTERS continue to interest subscribers. Dennis Sprague modified the p.45 program to write double size letters on command - the poke-ing is done automatically. Refer to the program on p. 45 and retain lines 9 thru 60, and line 120. Replace the rest with:

```

65 A = 20190
70 K = KP
80 IF K = 13 GOTO 100
90 K(A) = K; A = A+1; GOTO 70
100 K(A) = 0
105 CLEAR
110 CALL (B); GOTO 65

```

Dennis writes " 65 starts the display area, 100 shuts off the display if a zero is encountered. The ASCII values of K get poked into the display area 8 bits at a time."

With the above, enter and RUN the program. The just key in whatever letter, number, character that you wish to see, punch GO, and there it is, twice as big as life.

AMERICAN CONCERT FREQUENCIES chart has been prepared by Robert Hood, along with the closest Bally frequency: (all in Hertz)

Note	Standard	Bally	Note	Standard	Bally
C	261.7	262	G	392	392
C#	277.3	277	G#	415.2	415
D	293.7	294	A	440	440
Eb	311.1	311	Bb	466.1	466
E	329.7	330	B	493.9	494
F	349.2	349	C	523.3	524
F#	370.1	370			

In addition, Bob has furnished a program based on the equations of p.64 to solve for frequencies or tone register values, and this is found on p.70.

BANGMAN CORRECTION COMMENT by Ernie Sams indicates that perhaps Rory Wahl has a defective logic chip if Rory's correction in the last issue is needed to make the program work. Ernie writes:

"Rory suggests that the line should read:

2000 E=E+1; IF E=9 GOTO 9666; IF Q#1 GOSUB 9666+(EX16)

Q is a flag that is set to either 0 or 1. If it is set to 0 it sends the program to the man drawing routine at 9666, 9616, 9626, etc.

E is a counter that is to be incremented ONLY if the guess is wrong. It is NOT to be incremented if the guess is correct or if the letter has been previously used. So the portion of the program line, E=E+1, must follow the IF statement. Now, if the 'IF' portion of the IF E=9 GOTO 9666 statement is not satisfied the program defaults to the next numbered program line. The way Rory proposes, E would be incremented each time a guess is made, right or wrong. The program would never reach the man drawing routine statement because it can't go past the first IF statement until E=9 at which time the program goes to 9666, draws the gun and shoots the man that never gets drawn on the screen.

So line 2000 MUST remain exactly as was originally written or the program will not work as intended:

2000 IF Q#1 GOSUB 9666+(EX16); E=E+1; IF E=9 GOTO 9666

I have included all of Ernie's discourse as I felt that it would be of interest as a tutorial in why things are done in a certain way.

BOB HOOD's program to convert frequencies to register values and vice versa:

```

1 3. RETURN
4. FREQUENCIES
5. ROBERT HOOD
6. AUGUST 1979
7. NT=0
10. CLEAR; PRINT "DAILY
    TONE FREQUENCIES
20. PRINT "COMPUTES FREQU
    ENCY OF TONE
30. PRINT "REGISTER A B C
    R C IF VALUE OF
40. PRINT "MASTER & TONE
    REGISTERS
50. PRINT "ARE KNOWN. ALS
    O COMPUTES
60. PRINT "SETTINGS OF TON
    E REGISTER
70. PRINT "FOR A DESIRED
    FREQUENCY
80. PRINT "IF MASTER REGI
    STER VALUE
90. PRINT "IS KNOWN
100. PRINT "FOR FREQUENCY
    CALC INPUT 1
110. INPUT "FOR SETTING TO
    NE INPUT 2 ? "A
120. IF A=1 GOTO 150
130. IF A=2 GOTO 300
140. GOTO 100
150. CLEAR; INPUT "MASTER C
    OUNTER VALUE ? "M
160. INPUT "TONE COUNTER
    VALUE ? "T
170. F=100000/((M+1)*112))
    13 H=RM*10/((M+1)*112))

```

```

1172. G=100000/(T+1)); I=RM*
    10/(T+1)
1174. F=F*G+G*H/9+F*I/9+
    M*I/9
1190. PRINT; PRINT "FREQUENC
    Y IS ", F, "HERTZ"
1200. INPUT "INPUT 1 TO CON
    TINUE CALC. "Z
1210. IF Z=1 CLEAR; GOTO 100
    STOP
1220.
1300. CLEAR; INPUT "INPUT DE
    SIRED FREQUENCY ? "F
1310. R=100000/F; V=RM*10/F
1320. S=89
1330. PRINT; INPUT "SET MAST
    ER COUNTER VALUE ? "M
1340. S=R*S/(M+1)+V*S/(M
    +1)
1350. PRINT; PRINT "FOR FREQ
    UENCY OF ", F,
1360. PRINT "MASTER COUNTER
    ", M
1370. PRINT "TONE VALUE IS
    ", S
1380. PRINT; PRINT "INPUT 1
    TO CONTINUE
1390. INPUT "INPUT 2 TO STO
    P. ? "T
1400. IF T=1 CLEAR; GOTO 100
    STOP
1410.

```

TUTORIAL - SOUND SYNTHESIZER, Part 2 by Chuck Thomka

Whenever RESET is pushed, the &(16) to &(23) registers are set to fixed values. (This sort of thing is called DEFAULT) This also happens at POWER TURN ON. &(16) is set to 71, and &(17) through &(23) are set to zero.

Since pushing most keys on the keypad will generate a sound, one of the voices must be used. This means that since most keys have unique tones when pushed, they must be loading unique values into one or more of the registers. The voice used is the 'A' counter. Each key, when pushed, puts a value into the &(17) register that the 'A' counter will count up to. It will also put value 15 into the &(22) register, that will adjust the 'A' volume to its maximum so that the resultant frequency can be heard. At the end of the time of outputting the tone, the &(17) and &(22) are both put back to zero.

At anytime that the computer is stopped, the &(16) register will be set to 71, and &(17) and &(22) will be set to zero. This may affect some results of sound effects in programs where you want those registers to be left at some other values. All the other registers will be as they were last adjusted to, so remember this if you still have a tone or noise remaining after the computer has stopped.

Later in this article is a table of all the sound generating keys, their &(17) values, the resultant frequencies, and any special notes about them. (p.73)

The keys that do not generate sounds are $\frac{1}{2}$, x, +, and -. These keys will modify the sounds created by the other keys if the modifying keys are used just prior to the normal sound keys.

The divide key ($\frac{1}{2}$) will make the sound one octave lower in frequency than normal. This is done by temporarily making the master counter &(16) count twice as far. So while &(16) is normally at 71, for this one note it will be set to 143. As soon as the note has finished, &(16) will again return to 71 unless the next note is also preceded by a $\frac{1}{2}$.

The multiply key (x) will make the sound one octave higher in frequency. This is done by making &(16) equal 35 for the time the concerned note is sounding, at the end of which the &(16) will again return to 71.

The plus (+) and minus (-) keys are only used in conjunction with the numbers 1 through 7. This was arranged so that the plus and minus sign would be meaningful in playing musical sharps or flats in the Bally-mentioned 3 octave musical scale.

Another thing to mention is the "Note Timer" or NT. For each number of NT the notes played will be approximately 17 milliseconds long. An NT=0 results in no sound, while the maximum value of NT=255 results in about a 4.335 second note. ($0.017 \times 255 = 4.335$)

The 0 is used to extend the duration of a played note by taking the note timer and increasing it an additional NT quantity for each 0 following the note to be heard. For example, say we are to play a note while NT=10, and that this note is followed by 3 zeros, the resultant NT will be 40. After playing that modified NT, the NT will again return to normal (10 in this example) until called upon again.

A funny thing about this method of extending the duration of a played note is that you still cannot play any note longer than 4.335 seconds. This is because if you had a note timer extended by way of using zeros after a printed character, and it would result in an NT>255, the final result would probably be less than 255. To explain what I mean, you have to know about binary numbers and that the NT register is only 8 bits wide. If, for example, we had an NT of 50 and that some program that we are running is to print a character followed by 5 zeros, we would expect a temporary NT result of 300 ($1+5=6$, $6 \times 50=300$) but an 8 bit register's maximum bit count is only 255 while a binary conversion of

decimal 300 requires 9 bits (1 0010 1100). The result is that only the least 8 bits (0010 1100) will be loaded into the NT register, so NT will temporarily be 44. This you see is a lot shorter than we had at first expected and even shorter than the normal NT of 50.

TUTORIAL-SUBROUTINES

If you have a process that you want to have repeated a number of times, it is convenient and memory-saving to use the technique called SUBROUTINE, which requires the commands GOSUB and RETURN. I recently received a short program from Bret Dabel and Vince Garzoli that has this situation, and I thought that it might be of interest to all to show how a program can be modified this way. The program as it arrived is:

```
10 A=RND(32000)
20 INPUT "PLAYER #1 GUESS:" B
30 IF A=B PRINT B, "IS RIGHT"
40 IF A>B PRINT "MORE"
50 IF A<B PRINT "LESS"
60 INPUT "PLAYER #2 GUESS:" B
70 IF A=B PRINT B, "IS RIGHT"
80 IF A>B PRINT "MORE"
90 IF A<B PRINT "LESS"
100 INPUT "PLAYER #3 GUESS:" B
110 IF A=B PRINT B, "IS RIGHT"
120 IF A>B PRINT "MORE"
130 IF A<B PRINT "LESS"
140 INPUT "PLAYER #4 GUESS:" B
150 IF A=B PRINT B, "IS RIGHT"
160 IF A>B PRINT "MORE"
170 IF A<B PRINT "LESS"
180 IF A=B GOTO 10
190 GOTO 20
```

To utilize the SUBROUTINE command, we make the process to be repeated into a set of generalized statements and end them with the RETURN command. Then whenever you wish to perform the process, you direct the machine to the proper location with the GOSUB command, and when the machine does its job, it reads RETURN which tells it to go back to where it left the main program and pick up the next line number. This last statement is quite important.

As an example, let's review the Guessing Game program. We see that the A and B comparisons occur four times and so we can make a subroutine of them, giving them a set of line numbers away from the main program, as:

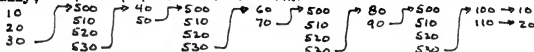
```
500 IF A = B PRINT B, "IS RIGHT"
510 IF A>B PRINT "MORE"
520 IF A<B PRINT "LESS"
530 RETURN
```

The program then reads:

```
10 A=RND(32000)
20 INPUT "PLAYER #1 GUESS:" B
30 GOSUB 500
40 INPUT "PLAYER #2 GUESS:" B
50 GOSUB 500
60 INPUT "PLAYER #3 GUESS:" B
70 GOSUB 500
80 INPUT "PLAYER #4 GUESS:" B
90 GOSUB 500
100 IF A=B GOTO 10
110 GOTO 20
500 IF A=B PRINT B, "IS RIGHT"
510 IF A>B PRINT "MORE"
520 IF A<B PRINT "LESS"
530 RETURN
```

If by chance you have written the subroutine at lines 500 - 530 but later you have added so much program that 500-530 will be buried in the program length, you will have a problem. As the computer completes line 490, it will search for the last value of A and B and perform the comparisons asked for in lines 500 through 520 (would you want that, then?) but it will HOW? when it gets to 530 because it is not in a subroutine mode and has no place to return to. To avoid this, you jump around the subroutine, in our case with a 490 GOTO 540. Since this is a legitimate operation, it means therefore that the subroutine could actually be placed anywhere within the program, and a suitable jump statement added.

Nesting of subroutines is possible. By this we mean that once you have gotten into the subroutine loop, you could have another subroutine called. The machine would perform the second subroutine and RETURN to the next line number of the first subroutine, and on its completion, go back to the main program. I believe that four such 'nests' are possible in the Bally BASIC, but one has to be very careful that each subroutine loop is completed - there cannot be any open loops. Diagrammatically, the guessing game looks like this:



RESULTANT FREQUENCY

NORMAL	÷ PREFIX	X PREFIX	CHARACTER(S)	B(17) VALUE
B(16) = 71	B(16) = 143	B(16) = 35		
54.63 Hz	27.32 Hz	109.26 Hz	!	37
57.97	28.98	115.93	"	34
61.43	30.71	122.85	#	32
64.98	32.49	129.97	\$	31
68.58	34.49	137.95	%	29
73.06	36.53	146.12	&	27
77.17	38.58	154.34	' (most)	26
81.77	40.88	163.53	(24
86.95	43.47	173.90)	23
92.14	46.07	184.28	*	21
102.89	51.45	205.78	, (comma)	20
115.39	57.70	230.78	. (period)	19
122.25	61.12	244.49	/	18
129.97	64.98	259.93	Y	17
137.19	68.59	274.37	Z	16
145.26	72.63	290.51	[+1 -2	15
154.34	77.17	308.67	\	14
164.62	82.31	328.25] +2 -3	13
173.90	86.95	347.80	↑ 3	11
184.28	92.14	368.56	← 4	10
195.48	97.99	391.96	↑ +4	9
205.78	102.89	411.56	→ 5	8
220.48	110.24	440.76	+5 -6	7
232.96	116.48	465.92	6 -7	6
246.94	123.47	493.87	7	5
262.70	131.35	525.40	+7 8	4
274.37	137.19	548.75	9	3
293.97	146.99	587.94	:	2
308.67	154.34	617.34	;	1

RESULTANT FREQUENCY

NORMAL	÷ PREFIX	X PREFIX	CHARACTER(S)	B(17) VALUE
B(16) = 71	B(16) = 143	B(16) = 35		
324.92 Hz	162.46 Hz	649.83 Hz	<	37
352.77	176.38	705.53	=	34
374.15	187.07	748.29	>	32
385.84	192.72	771.68	?	31
411.56	205.78	823.12	@	29
440.96	220.48	881.91	A	27
457.29	228.64	914.58	B	26
493.87	246.94	987.74	C	24
514.45	257.23	1028.90	D	23
561.22	280.61	1122.44	E	21
587.94	293.97	1175.89	F	20
617.34	308.67	1234.68	G	19
649.83	324.92	1297.66	H	18
685.93	342.97	1371.87	I	17
726.28	363.14	1452.57	J	16
771.68	385.84	1543.35	K	15
823.12	411.56	1646.24	L	14
881.91	440.76	1763.83	M	13
1028.90	514.45	2057.80	N	11
1122.44	561.22	2244.87	O	10
1234.68	617.34	2469.36	P	9
1371.87	685.93	2743.73	Q	8
1543.35	771.68	3086.70	R	7
1763.83	881.91	3527.66	S	6
2057.80	1028.90	4115.60	T	5
2469.36	1234.68	4938.72	U	4
3086.70	1543.35	6173.40	V	3
4115.60	2057.80	8231.20	W	2
6173.40	3086.70	12346.81	X	1

Line #	Statements()	Line #	Statements()
1	BLACK BOX	280	FC=9;GOTO 250
2	BY B. REANY	300	CX=-77;CY=40;INPUT C
10	BC=1A	310	CX=-41;CY=40;INPUT R
15	BC=BC-1.6;FC=0	315	C=12-C-2.9;R=24-2*(R-10)
20	CLEAR	320	BOX C,R;3,3
25	@(1)=2	325	CY=40;PRINT "
30	@(2)=2	330	NT=3;GOTO 250
35	NT=50;CX=-41	400	CY=40;CX=-77;INPUT R
40	PRINT "BLACK BOX	405	NT=0;@ (2) = @ (2) + 1
50	CX=-47;CY=0;NT=0	410	B=R+10;C=R+1
55	PRINT "HOW MANY ATOMS	415	IF B=0 1-C;K=0;L=0;M=1
60	CX=-41	420	IF B=1 1-C;K=C;L=1;M=0
65	PRINT "DO YOU WANT?	425	IF B=2 1-C;K=9;L=0;M=1
70	CX=-32;INPUT A	430	IF B=3 1-C;K=C;L=1;M=0
100	FOR B=1 TO A	450	CX=-77;CY=40;D=10;J=K
105	C=RND(8)+10;RND(8)	460	IF @ (3) = 1 PRINT "ABSORBED";
110	IF @ (C) = 1 B=B-1	465	GOTO 325
115	@ (C) = 1	470	IF L=0 S=D+10;M=T+S-20
120	NEXT B	475	IF M=0 S=D+10;L=T+S-2
125	CLEAR;CY=32	480	IF T=1 T=1
135	CX=-29;PRINT "0 1 2 3 4	485	IF @ (S) = 1 IF @ (T) = 1 L=L;
140	FOR B=10 TO 17	490	M=M;GOTO 500
145	CX=-47;PRINT "0 1 2 3 4	500	IF L=0 IF @ (S) = 1 L=L;M=M;
150	NEXT B	505	GOTO 500
155	CX=-35;PRINT "0 1 2 3 4	510	IF L=0 IF @ (T) = 1 L=L;
160	FOR B=10 TO 17	515	M=M;GOTO 500
165	CX=-47;PRINT "0 1 2 3 4	520	IF M=0 IF @ (T) = 1 M=M;L=L;
170	NEXT B	525	GOTO 500
175	CX=-35;PRINT "0 1 2 3 4	530	IF M=0 IF @ (S) = 1 M=M;L=L;
180	FOR B=10 TO 17	535	GOTO 500
185	CX=-47;PRINT "0 1 2 3 4	540	IF M=0 IF @ (T) = 1 M=M;L=L;
190	NEXT B	545	GOTO 500
195	CX=-35;PRINT "0 1 2 3 4	550	IF M=0 IF @ (S) = 1 M=M;L=L;
200	FOR B=10 TO 17	555	GOTO 500
205	CX=-47;PRINT "0 1 2 3 4	560	IF M=0 IF @ (T) = 1 M=M;L=L;
210	NEXT B	565	GOTO 500
215	CX=-35;PRINT "0 1 2 3 4	570	IF M=0 IF @ (S) = 1 M=M;L=L;
220	FOR B=10 TO 17	575	GOTO 500
225	CX=-47;PRINT "0 1 2 3 4	580	IF M=0 IF @ (T) = 1 M=M;L=L;
230	NEXT B	585	GOTO 500
235	CX=-35;PRINT "0 1 2 3 4	590	IF M=0 IF @ (S) = 1 M=M;L=L;
240	FOR B=10 TO 17	595	GOTO 500
245	CX=-47;PRINT "0 1 2 3 4	600	IF M=0 IF @ (T) = 1 M=M;L=L;
250	NEXT B	605	GOTO 500
255	CX=-35;PRINT "0 1 2 3 4	610	IF M=0 IF @ (S) = 1 M=M;L=L;
260	FOR B=10 TO 17	615	GOTO 500
265	CX=-47;PRINT "0 1 2 3 4	620	IF M=0 IF @ (T) = 1 M=M;L=L;
270	NEXT B	625	GOTO 500
275	CX=-35;PRINT "0 1 2 3 4	630	IF M=0 IF @ (S) = 1 M=M;L=L;
280	FOR B=10 TO 17	635	GOTO 500
285	CX=-47;PRINT "0 1 2 3 4	640	IF M=0 IF @ (T) = 1 M=M;L=L;
290	NEXT B	645	GOTO 500
295	CX=-35;PRINT "0 1 2 3 4	650	IF M=0 IF @ (S) = 1 M=M;L=L;
300	FOR B=10 TO 17	655	GOTO 500
305	CX=-47;PRINT "0 1 2 3 4	660	IF M=0 IF @ (T) = 1 M=M;L=L;
310	NEXT B	665	GOTO 500
315	CX=-35;PRINT "0 1 2 3 4	670	IF M=0 IF @ (S) = 1 M=M;L=L;
320	FOR B=10 TO 17	675	GOTO 500
325	CX=-47;PRINT "0 1 2 3 4	680	IF M=0 IF @ (T) = 1 M=M;L=L;
330	NEXT B	685	GOTO 500
335	CX=-35;PRINT "0 1 2 3 4	690	IF M=0 IF @ (S) = 1 M=M;L=L;
340	FOR B=10 TO 17	695	GOTO 500
345	CX=-47;PRINT "0 1 2 3 4	700	IF M=0 IF @ (T) = 1 M=M;L=L;
350	NEXT B	705	GOTO 500
355	CX=-35;PRINT "0 1 2 3 4	710	IF M=0 IF @ (S) = 1 M=M;L=L;
360	FOR B=10 TO 17	715	GOTO 500
365	CX=-47;PRINT "0 1 2 3 4	720	IF M=0 IF @ (T) = 1 M=M;L=L;
370	NEXT B	725	GOTO 500
375	CX=-35;PRINT "0 1 2 3 4	730	IF M=0 IF @ (S) = 1 M=M;L=L;
380	FOR B=10 TO 17	735	GOTO 500
385	CX=-47;PRINT "0 1 2 3 4	740	IF M=0 IF @ (T) = 1 M=M;L=L;
390	NEXT B	745	GOTO 500
395	CX=-35;PRINT "0 1 2 3 4	750	IF M=0 IF @ (S) = 1 M=M;L=L;
400	FOR B=10 TO 17	755	GOTO 500
405	CX=-47;PRINT "0 1 2 3 4	760	IF M=0 IF @ (T) = 1 M=M;L=L;
410	NEXT B	765	GOTO 500
415	CX=-35;PRINT "0 1 2 3 4	770	IF M=0 IF @ (S) = 1 M=M;L=L;
420	FOR B=10 TO 17	775	GOTO 500
425	CX=-47;PRINT "0 1 2 3 4	780	IF M=0 IF @ (T) = 1 M=M;L=L;
430	NEXT B	785	GOTO 500
435	CX=-35;PRINT "0 1 2 3 4	790	IF M=0 IF @ (S) = 1 M=M;L=L;
440	FOR B=10 TO 17	795	GOTO 500
445	CX=-47;PRINT "0 1 2 3 4	800	IF M=0 IF @ (T) = 1 M=M;L=L;
450	NEXT B	805	GOTO 500
455	CX=-35;PRINT "0 1 2 3 4	810	IF M=0 IF @ (S) = 1 M=M;L=L;
460	FOR B=10 TO 17	815	GOTO 500
465	CX=-47;PRINT "0 1 2 3 4	820	IF M=0 IF @ (T) = 1 M=M;L=L;
470	NEXT B	825	GOTO 500
475	CX=-35;PRINT "0 1 2 3 4	830	IF M=0 IF @ (S) = 1 M=M;L=L;
480	FOR B=10 TO 17	835	GOTO 500
485	CX=-47;PRINT "0 1 2 3 4	840	IF M=0 IF @ (T) = 1 M=M;L=L;
490	NEXT B	845	GOTO 500
495	CX=-35;PRINT "0 1 2 3 4	850	IF M=0 IF @ (S) = 1 M=M;L=L;
500	FOR B=10 TO 17	855	GOTO 500
505	CX=-47;PRINT "0 1 2 3 4	860	IF M=0 IF @ (T) = 1 M=M;L=L;
510	NEXT B	865	GOTO 500
515	CX=-35;PRINT "0 1 2 3 4	870	IF M=0 IF @ (S) = 1 M=M;L=L;
520	FOR B=10 TO 17	875	GOTO 500
525	CX=-47;PRINT "0 1 2 3 4	880	IF M=0 IF @ (T) = 1 M=M;L=L;
530	NEXT B	885	GOTO 500
535	CX=-35;PRINT "0 1 2 3 4	890	IF M=0 IF @ (S) = 1 M=M;L=L;
540	FOR B=10 TO 17	895	GOTO 500
545	CX=-47;PRINT "0 1 2 3 4	900	IF M=0 IF @ (T) = 1 M=M;L=L;
550	NEXT B	905	GOTO 500
555	CX=-35;PRINT "0 1 2 3 4	910	IF M=0 IF @ (S) = 1 M=M;L=L;
560	FOR B=10 TO 17	915	GOTO 500
565	CX=-47;PRINT "0 1 2 3 4	920	IF M=0 IF @ (T) = 1 M=M;L=L;
570	NEXT B	925	GOTO 500
575	CX=-35;PRINT "0 1 2 3 4	930	IF M=0 IF @ (S) = 1 M=M;L=L;
580	FOR B=10 TO 17	935	GOTO 500
585	CX=-47;PRINT "0 1 2 3 4	940	IF M=0 IF @ (T) = 1 M=M;L=L;
590	NEXT B	945	GOTO 500
595	CX=-35;PRINT "0 1 2 3 4	950	IF M=0 IF @ (S) = 1 M=M;L=L;
600	FOR B=10 TO 17	955	GOTO 500
605	CX=-47;PRINT "0 1 2 3 4	960	IF M=0 IF @ (T) = 1 M=M;L=L;
610	NEXT B	965	GOTO 500
615	CX=-35;PRINT "0 1 2 3 4	970	IF M=0 IF @ (S) = 1 M=M;L=L;
620	FOR B=10 TO 17	975	GOTO 500
625	CX=-47;PRINT "0 1 2 3 4	980	IF M=0 IF @ (T) = 1 M=M;L=L;
630	NEXT B	985	GOTO 500
635	CX=-35;PRINT "0 1 2 3 4	990	IF M=0 IF @ (S) = 1 M=M;L=L;
640	FOR B=10 TO 17	995	GOTO 500
645	CX=-47;PRINT "0 1 2 3 4	1000	IF M=0 IF @ (T) = 1 M=M;L=L;
650	NEXT B	1005	GOTO 500
655	CX=-35;PRINT "0 1 2 3 4	1010	IF M=0 IF @ (S) = 1 M=M;L=L;
660	FOR B=10 TO 17	1015	GOTO 500
665	CX=-47;PRINT "0 1 2 3 4	1020	IF M=0 IF @ (T) = 1 M=M;L=L;
670	NEXT B	1025	GOTO 500
675	CX=-35;PRINT "0 1 2 3 4	1030	IF M=0 IF @ (S) = 1 M=M;L=L;
680	FOR B=10 TO 17	1035	GOTO 500
685	CX=-47;PRINT "0 1 2 3 4	1040	IF M=0 IF @ (T) = 1 M=M;L=L;
690	NEXT B	1045	GOTO 500
695	CX=-35;PRINT "0 1 2 3 4	1050	IF M=0 IF @ (S) = 1 M=M;L=L;
700	FOR B=10 TO 17	1055	GOTO 500
705	CX=-47;PRINT "0 1 2 3 4	1060	IF M=0 IF @ (T) = 1 M=M;L=L;
710	NEXT B	1065	GOTO 500
715	CX=-35;PRINT "0 1 2 3 4	1070	IF M=0 IF @ (S) = 1 M=M;L=L;
720	FOR B=10 TO 17	1075	GOTO 500
725	CX=-47;PRINT "0 1 2 3 4	1080	IF M=0 IF @ (T) = 1 M=M;L=L;
730	NEXT B	1085	GOTO 500
735	CX=-35;PRINT "0 1 2 3 4	1090	IF M=0 IF @ (S) = 1 M=M;L=L;
740	FOR B=10 TO 17	1095	GOTO 500
745	CX=-47;PRINT "0 1 2 3 4	1100	IF M=0 IF @ (T) = 1 M=M;L=L;
750	NEXT B	1105	GOTO 500
755	CX=-35;PRINT "0 1 2 3 4	1110	IF M=0 IF @ (S) = 1 M=M;L=L;
760	FOR B=10 TO 17	1115	GOTO 500
765	CX=-47;PRINT "0 1 2 3 4	1120	IF M=0 IF @ (T) = 1 M=M;L=L;
770	NEXT B	1125	GOTO 500
775	CX=-35;PRINT "0 1 2 3 4	1130	IF M=0 IF @ (S) = 1 M=M;L=L;
780	FOR B=10 TO 17	1135	GOTO 500
785	CX=-47;PRINT "0 1 2 3 4	1140	IF M=0 IF @ (T) = 1 M=M;L=L;
790	NEXT B	1145	GOTO 500
795	CX=-35;PRINT "0 1 2 3 4	1150	IF M=0 IF @ (S) = 1 M=M;L=L;
800	FOR B=10 TO 17	1155	GOTO 500
805	CX=-47;PRINT "0 1 2 3 4	1160	IF M=0 IF @ (T) = 1 M=M;L=L;
810	NEXT B	1165	GOTO 500
815	CX=-35;PRINT "0 1 2 3 4	1170	IF M=0 IF @ (S) = 1 M=M;L=L;
820	FOR B=10 TO 17	1175	GOTO 500
825	CX=-47;PRINT "0 1 2 3 4	1180	IF M=0 IF @ (T) = 1 M=M;L=L;
830	NEXT B	1185	GOTO 500
835	CX=-35;PRINT "0 1 2 3 4	1190	IF M=0 IF @ (S) = 1 M=M;L=L;
840	FOR B=10 TO 17	1195	GOTO 500
845	CX=-47;PRINT "0 1 2 3 4	1200	IF M=0 IF @ (T) = 1 M=M;L=L;
850	NEXT B	1205	GOTO 500
855	CX=-35;PRINT "0 1 2 3 4	1210	IF M=0 IF @ (S) = 1 M=M;L=L;
860	FOR B=10 TO 17	1215	GOTO 500
865	CX=-47;PRINT "0 1 2 3 4	1220	IF M=0 IF @ (T) = 1 M=M;L=L;
870	NEXT B	1225	GOTO 500
875	CX=-35;PRINT "0 1 2 3 4	1230	IF M=0 IF @ (S) = 1 M=M;L=L;
880	FOR B=10 TO 17	1235	GOTO 500
885	CX=-47;PRINT "0 1 2 3 4	1240	IF M=0 IF @ (T) = 1 M=M;L=L;
890	NEXT B	1245	GOTO 500
895	CX=-35;PRINT "0 1 2 3 4	1250	IF M=0 IF @ (S) = 1 M=M;L=L;
900	FOR B=10 TO 17	1255	GOTO 500
905	CX=-47;PRINT "0 1 2 3 4	1260	IF M=0 IF @ (T) = 1 M=M;L=L;
910	NEXT B	1265	GOTO 500
915	CX=-35;PRINT "0 1 2 3 4	1270	IF M=0 IF @ (S) = 1 M=M;L=L;
920	FOR B=10 TO 17	1275	GOTO 500
925	CX=-47;PRINT "0 1 2 3 4	1280	IF M=0 IF @ (T) = 1 M=M;L=L;
930	NEXT B	1285	GOTO 500
935	CX=-35;PRINT "0 1 2 3 4	1290	IF M=0 IF @ (S) = 1 M=M;L=L;
940	FOR B=10 TO 17	1295	GOTO 500
945	CX=-47;PRINT "0 1 2 3 4	1300	IF M=0 IF @ (T) = 1 M=M;L=L;
950	NEXT B	1305	GOTO 500
955	CX=-35;PRINT "0 1 2 3 4	1310	IF M=0 IF @ (S) = 1 M=M;L=L;
960	FOR B=10 TO 17	1315	GOTO 500
965	CX=-47;PRINT "0 1 2 3 4	1320	IF M=0 IF @ (T) = 1 M=M;L=L;
970	NEXT B	1325	GOTO 500
975	CX=-35;PRINT "0 1 2 3 4	1330	IF M=0 IF @ (S) = 1 M=M;L=L;
980	FOR B=10 TO 17	1335	GOTO 500
985	CX=-47;PRINT "0 1 2 3 4	1340	IF M=0 IF @ (T) = 1 M=M;L=L;
990	NEXT B	1345	GOTO 500
995	CX=-35;PRINT "0 1 2 3 4	1350	IF M=0 IF @ (S) = 1 M=M;L=L;
1000	FOR B=10 TO 17	1355	GOTO 500
1005	CX=-47;PRINT "0 1 2 3 4	1360	IF M=0 IF @ (T) = 1 M=M;L=L;
1010	NEXT B	1365	GOTO 500
1015	CX=-35;PRINT "0 1 2 3 4	1370	IF M=0 IF @ (S) = 1 M=M;L=L;
1020	FOR B=10 TO 17	1375	GOTO 500
1025	CX=-47;PRINT "0 1 2 3 4	1380	IF M=0 IF @ (T) = 1 M=M;L=L;
1030	NEXT B	1385	GOTO 500
1035	CX=-35;PRINT "0 1 2 3 4	1390	IF M=0 IF @ (S) = 1 M=M;L=L;
1040	FOR B=10 TO 17	1395	GOTO 500
1045	CX=-47;PRINT "0 1 2 3 4	1400	IF M=0 IF @ (T) = 1 M=M;L=L;
1050	NEXT B	1405	GOTO 500
1055	CX=-35;PRINT "0 1 2 3 4	1410	IF M=0 IF @ (S) = 1 M=M;L=L;
1060	FOR B=10 TO 17	1415	GOTO 500
1065	CX=-47;PRINT "0 1 2 3 4	1420	IF M=0 IF @ (T) = 1 M=M;L=L;
1070	NEXT B	1425	GOTO 500
1075	CX=-35;PRINT "0 1 2 3 4	1430	IF M=0 IF @ (S) = 1 M=M;L=L;
1080	FOR B=10 TO 17	1435	GOTO 500
1085	CX=-47;PRINT "0 1 2 3 4	1440	IF M=0 IF @ (T) = 1 M=M;L=L;
1090	NEXT B	1445	GOTO 500
1095	CX=-35;PRINT "0 1 2 3 4	1450	IF M=0 IF @ (S) = 1 M=M;L=L;
1100	FOR B=10 TO 17	1455	GOTO 500
1105	CX=-47;PRINT "0 1 2 3 4	1460	IF M=0 IF @ (T) = 1 M=M;L=L;
1110	NEXT B	1465	GOTO 500
1115	CX=-35;PRINT "0 1 2 3 4	1470	IF M=0 IF @ (S) = 1 M=M;L=L;
1120	FOR B=10 TO 17	1475	GOTO 500
1125	CX=-47;PRINT "0 1 2 3 4	1480	IF M=0 IF @ (T) = 1 M=M;L=L;
1130	NEXT B	1485	GOTO 500
1135	CX=-35;PRINT "0 1 2 3 4	1490	IF M=0 IF @ (S) = 1 M=M;L=L;
1140	FOR B=10 TO 17	1495	GOTO 500
1145	CX=-47;PRINT "0 1 2 3 4	1500	IF M=0 IF @ (T) = 1 M=M;L=L;
1150	NEXT B	1505	GOTO 500
1155	CX=-35;PRINT "0 1 2 3 4	1510	IF M=0 IF @ (S) = 1 M=M;L=L;
1160	FOR B=10 TO 17	1515	GOTO 500
1165	CX=-47;PRINT "0 1 2 3 4	1520	IF M=0 IF @ (T) = 1 M=M;L=L;
1170	NEXT B	1525	GOTO 500
1175	CX=-35;PRINT "0 1 2 3 4	1530	IF M=0 IF @ (S) = 1 M=M;L=L;
1180	FOR B=10 TO 17	1535	GOTO 500
1185	CX=-47;PRINT "0 1 2 3 4	1540	IF M=0 IF @ (T) = 1 M=M;L=L;
1190	NEXT B	1	

arcadian

POKE-ING PROGRAM allows you to load machine instructions into the @ string, which means that you can call several machine language subroutines from inside the BASIC. Developed by George Breadon, the program follows along with some data to be inserted that will call up our old buddy, ARCADIAN (ref.p.45)

```

5 NT = 0
10 A = 20180; B=A; For K=0 TO 13 } INPUT MACHINE INSTRUCTIONS
20 INPUT @(K); NEXT K } INTO @ STRING
30 FOR K = 0 TO 13; CLEAR
40 CY = 0; PRINT K, @(K)
50 D = KP; IF D=31 GOTO 80
60 IF D=57 GOTO 90
70 GOTO 50
80 INPUT "CHANGE="; L; @(K)=L
90 NEXT K
100 A=B; FOR K= 0 TO 13 } POKE @STRING INTO MEMORY
110 @(A)=@(K); A=A+2; NEXT K
120 IF @(20) = 8 GOTO 30
130 C=20180; GOSUB 160
140 C=20190; GOSUB 160 } HIT "GOTO" KEY TO BRANCH
150 GOTO 120 } BACK TO EDIT ROUTINE AT
160 CLEAR; CALL (C); RETURN } ANY TIME

```

DATA to be inserted: This is all in machine level code.

@(0) =	-43	@(7) =	27672	
1	12341	8	20200	
2	19480	9	-13871	@ 0 thru 4 go into 20180
3	3164 Or 3159	10	21057	while @ 5 thru 13 go into
4	-13871	11	16707	20190, two at a time
5	-43	12	18756	
6	53	13	20033	

SOFTWARE PRODUCERS are invited to contact VIDEO CONCEPTS at 625 W. 53 Ave, Anchorage Alaska, 99502, for distribution of their products thru the store up in the cold country.

RETURNED BALLY UNITS are available from V. Jupe, Star Route Box 60, Carlotta, CA, 95528 These are working, and at less than \$200. Also some games, write.

ADS start here this time:

SELL ARCADE with 4 controls, BASIC cassette and interface. BLACKJACK, BASEBALL, RED BARON, CROSWORDS Interface has Jack for printer. Reasonable Offer to Bob Schwind 12311 W. Silver Spring Dr. Milwaukee, WI 53225 (414) 367-4804

ADS

SELL ARCADE complete, includes PANZER ATTACK,CLOWNS,ESCAPE,FOOTBALL, BASEBALL,BINGO MATH,LETTER MATCH, PLACKJACK,ETC., BASIC and CASSETTE INTERFACE. Total original list price 560. First certified check for 400, or best offer. B.PERLSON 6400 N.EIM TREE RD. MILWAUKEE WI, 53217 414-352-1331

Quality games on C-10 cassettes: STARELASTER (2 player spacewar) and HAMMURABI (you control ancient Sumeria) at 7.each, both for 12.
Dan. Pierce 229 Orville St Apt 1 Fairborn OH 45324

LISTING for the game SUB SEARCH, a one-player item, at \$1.25
Marc Gladstein 1213½ S. ALFRED ST Los Angeles CA 90035 (213)658-5804

Available through Sebree's Computing 456 Granite Ave., Monrovia CA 91016- Games:3.95-UFO BATTLE, HIT THE PEDESTRIAN, SUBMARINE MINEFIELD; 2.95-MUNCH!;5.50-DOWN THE TRENCH;\$8.95-***SUPER WUMPUS**;\$2.50-MATH ROUTINES (calculates Sine, Cosine, Arctangent, & Square Root!!). All programs with one page of documentation/instructions. Send for descriptions. Timothy Hays.

A note from W&W Software that they have another cassette ready.

SELL Bally ARCADE EPA 1100 with BASIC,FOOTBALL,BASEBALL, 4 other cassettes, tape interface \$275. Geo. Evanoff, 10028 N.E. 28th Place, Bellevue WA 98004 (206)-827-2918

One player game called SUBSEARCH, 1.25 for listing, only. Marc Gladstein 1213½ S. Alfred St. Los Angeles CA 90035 (213) 658-5804

REVIEW of programs has been suggested by some subscribers, who are concerned about purchasing a 'pig in a poke'. IF someone else is willing to do a critical review of a program that some advertiser is also willing to submit, I will get the two parties together and accept the review for publication. The opinions will be the reviewers, not mine.

=76=

ARCADIAN

Robert Fabris, stamp licker
3626 Morrie Dr.
San José, CA 95127

FIRST CLASS


```

10 A = RND (32000)
20 INPUT "PLAYER #1 GUESS:" B
30 IF A=B PRINT "MORE"
40 IF A > B PRINT "LESS"
50 INPUT "PLAYER #2 GUESS:" B
60

```

```

10 A=RND(32000)
20 INPUT "PLAYER #1 GUESS:" B
30 IF A=B PRINT B, "IS RIGHT"
40 IF A>B PRINT "MORE"
50 IF A<B PRINT "LESS"
60 INPUT "PLAYER #2 GUESS:" B
70 IF A=B PRINT B, "IS RIGHT"
80 IF A>B PRINT "MORE"
90 IF A<B PRINT "LESS"
100 INPUT "PLAYER #3 GUESS:" B
110 IF A=B PRINT B, "IS RIGHT"
120 IF A>B PRINT "MORE"
130 IF A<B PRINT "LESS"
140 INPUT "PLAYER #4 GUESS:" B
150 IF A=B PRINT B, "IS RIGHT"
160 IF A>B PRINT "MORE"
170 IF A<B PRINT "LESS"
180 IF A=B GOTO 10
190 GOTO 20

```

```

10 A=RND(32000)
20 INPUT "PLAYER #1 GUESS:" B
30 GOSUB 500
40 INPUT "PLAYER #2 GUESS:" B
50 GOSUB 500
60 INPUT "PLAYER #3 GUESS:" B
70 GOSUB 500
80 INPUT "PLAYER #4 GUESS:" B
90 GOSUB 500
100 IF A=B GOTO 10
110 GOTO 20
500 IF A=B PRINT B, "IS RIGHT"
510 IF A>B PRINT "MORE"
520 IF A<B PRINT "LESS"
530 RETURN

```

for 71

BLACK BOX RULES:

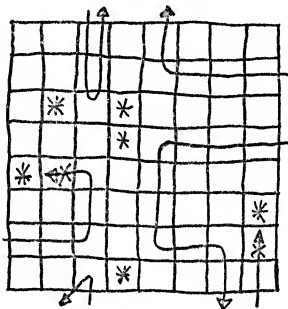
An 8 by 8 grid has a predetermined number of atoms hidden, one per square, under the grid squares. Berthold rays will be generated after you select a ray entry point after pressing the "1" key. Rays travel in straight lines perpendicular to the grid edge, starting from the ray entry point, until they are absorbed or exit from the grid. They obey the following rules;

1. A ray entering the grid on either side of an atom on the edge of the grid is deflected backward and away from the edge atom.
 2. A ray aimed between two atoms with an open square between them is reflected back upon its' path.
 3. A ray coming within one square diagonally of an atom is deflected away 90 degrees from that atom.
 4. A ray colliding with an atom will be absorbed, and its' absorption will be signalled upon the screen
 5. A ray emerging from the grid will signal its' exit point.
- The "2" key will ask for a col (0 to 7) and row (10 to 17) and will either mark or unmark the grid position at their intersection where you suspect an atom is located. The "3" key will display the grid points where the atoms were located, those which you marked as having atoms, and will grade your guesses, and clear the grid for the next game. The zero key will restart the game, but if the grid was not cleared with the "3" key, the preceding grid atoms will not be cleared.

After this program is loaded, the direct executed "PRINT SZ" command must print at least 200, or the program strings will be insufficient to execute. For this reason, closing quote marks on literals, as well as several obvious input edits, have been deleted.

On the listing, spaces have been added to separate necessary line elements from each other; asterisks replace the multiply "x", and the slash replaces the divide sign, thanks to a limited typewriter keyboard. At least, the 's' key is fixed, so I don't have to lithp on it anymore.

This program is unconditionally guaranteed by the author to be smack up against your core limitation, or double your core dumps back.



Thank You
B. Reany
1106 E. Julia Dr.
Perry, Fl 32347

9

```

1 .
2 .
3 .
10 BC=14
15 BC=BC+16;FC=0
20 CLEAR
25 @ (1)=2
30 @ (2)=2
35 NT=50;CX=-41
45 PRINT " BLACK BOX
50 CX=-47;CY=0;NT=0
55 PRINT "HOW MANY ATOMS
60 CX=-41
65 PRINT "DO YOU WANT?
70 CY=-32;INPUT A
100 FOR B=1 TO A
105 C=RND(8)+10*RND(8)
110 IF @ (C)=1 B=B-1
115 @ (C)=1
120 NEXT B
125 CLEAR;CY=32
135 CX=-29;PRINT "0 1 2 3 4 5 6 7
140 CX=-47;PRINT "10";CX=67;PRINT "30
145 CX=-47;PRINT "11";CX=67;PRINT "31
150 CX=-47;PRINT "12";CX=67;PRINT "32
155 CX=-47;PRINT "13";CX=67;PRINT "33
160 CX=-47;PRINT "14";CX=67;PRINT "34
165 CX=-47;PRINT "15";CX=67;PRINT "35
170 CX=-47;PRINT "16";CX=67;PRINT "36
175 CX=-47;PRINT "17";CX=67;PRINT "37
180 CX=-35;PRINT "20 1 2 3 4 5 6 7";CY=40
185 BOX 13,-4,97,65,3
190 FOR B=-29 TO 55 STEP 12
195 FOR C=24 TO -32 STEP -8
200 BOX B,C,11,7,3
205 NEXT C
210 NEXT B
250 FOR B=1 TO 2400
255 IF @ (22)=16 GOTO 15
260 IF @ (23)=8 GOTO 400
265 IF @ (22)=8 GOTO 300
270 IF @ (21)=8 GOTO 800
275 NEXT B
280 FC=9;GOTO 250
300 CX=-77;CY=40;INPUT C
310 CX=-41;CY=40;INPUT R
315 C=12*C-29;R=24-8*(R-10)
320 BOX C,R,3,3,3
325 CY=40;PRINT "
330 NT=3;GOTO 250

400 CY=40;CX=-77;INPUT R
405 NT=0;@ (2)=@ (2)+1
410 B=R/10;C=RM+1
415 IF B=0 J=C;K=0;L=0;M=1
420 IF B=1 J=0;K=C;L=1;M=0
425 IF B=2 J=C;K=9;L=0;M=-1
430 IF B=3 J=9;K=C;L=-1;M=0
450 CX=-77;CY=40;D=10*J+K
460 IF @ (D)=1 PRINT "ABSORBED";
      GOTO 325
465 IF L=0 S=D+10+M;T=S-20
470 IF M=0 S=D+1+10*L;T=S-2
471 IF T<1 T=1
475 IF @ (S)=1 IF @ (T)=1 L=-L;
      M=-M;GOTO 500
480 IF L=0 IF @ (S)=1 L=-1;M=0;
      GOTO 500
485 IF L=0 IF @ (T)=1 L=1;M=0;
      GOTO 500
490 IF M=0 IF @ (S)=1 M=-1;L=0;
      GOTO 500
495 IF M=0 IF @ (T)=1 M=1;L=0
500 J=J+1;K=K+M
505 IF J<1 GOTO 550
510 IF J>8 GOTO 550
515 IF K<1 GOTO 550
520 IF K>8 GOTO 550
525 GOTO 450
550 K=32-8*K;J=12*J-41
560 NT=50
565 FOR B=1 TO 50
570 BOX J,K,11,7,3
575 NEXT B
580 NT=3;GOTO 325
800 FOR B=11 TO 88
805 C=B/10-1;D=RM-1
820 C=-29+C*12;D=24-D*8
830 IF @ (B)=1 BOX C,D,7,5,3;@ (B)=0
840 IF PX(C,D)=1 @ (1)=@ (1)+1
850 NEXT B
855 NT=50;CY=40;CX=-35
865 PRINT "F I N I S H
870 NT=3;CLEAR
875 IF @ (1)>10 GOTO 900
880 PRINT "EUREKA!
885 CY=0;PRINT "YOU HAD ",#2,A,"ATOMS
890 PRINT "YOU USED ",#1,@ (2)-2," RATS
895 GOTO 250
900 PRINT " SORRY
910 GOTO 250

```

```

140 FOR B=10 TO 17
145 CX=-47;PRINT #2,B,3;
      CX=67;PRINT #2,B+20
150 NEXT B

```

B. REANY
32847

11

Try

411 IF B > 3 GOTO 400

412 IF C > 8 GOTO 400

305 IF C > 7 GOTO 300

312 IF R < 10 GOTO 380

313 IF R > 17 GOTO 310

IF you have room

```

1 .BLACK BOX
2 .BY B. REANY
10 BC=1A
15 BC=BC+16; FC=φ
20 CLEAR
25 @(1)=2
30 @(2)=2
35 NT=5φ; CX=-4.1
40 PRINT" BLACK BOX
50 CX=-47; CY=φ; NT=φ
55 PRINT" HOW MANY ATOMS
60 CX=-4.1
65 PRINT" DO YOU WANT?
70 CY=-32; INPUT A
100 FOR B=1 TO A
105 C=RND(8)+1φ×RND(8)
110 IF@(C)=1 B=B-1
115 @(C)=1
120 NEXT B
125 CLEAR; CY=32
135 CX=-29; PRINT" φ 1 2 3 4
    5 6 7
140 FOR B=1φ TO 17
145 CX=-47; PRINT#2,B; CX=67;
    PRINT#2,B+2φ
150 NEXT B
180 CX=-35; PRINT" 2φ 1 2 3 4
    5 6 7'; CY=4φ
185 BOX 13,-4,97,65,3
190 FOR B=-29 TO 55 STEP 12
195 FOR C=24 TO -32 STEP -8
200 BOX B,C,11,7,3
205 NEXT C
210 NEXT B
250 FOR B=1 TO 24φφ
255 IF&(22)=16 GOTO 15
260 IF&(23)=8 GOTO 4φφ
265 IF&(22)=8 GOTO 3φφ
270 IF&(21)=8 GOTO 8φφ
275 NEXT B

```

Line #	Statement(s)	Comments
280	FC=9; GOTO 250	
300	CX=-77; CY=40; INPUT C	
310	CX=-41; CY=40; INPUT R	
315	C=12xC-29; R=24-8x(R-10)	
320	BOX C, R, 3, 3, 3	
325	CY=40; PRINT " " ,	
330	NT=3; GOTO 250	
400	CY=40; CX=-77; INPUT R	
405	NT=0; @ (2) = @ (2) + 1	
410	B=R÷10; C=RM+1	
415	IF B=0 J=C; K=0; L=0; M=1	
420	IF B=1 J=0; K=C; L=1; M=0	
425	IF B=2 J=C; K=9; L=0; M=-1	
430	IF B=3 J=9; K=C; L=-1; M=0	
450	CX=-77; CY=40; D=10xJ+K	
460	IF @ (D) = 1 PRINT "ABSORBED"; GOTO 325	
465	IF L=0 S=D+10+M; T=S-20	
470	IF M=0 S=D+1+10xL; T=S-2	
471	IF T<1 T=1	
475	IF @ (S) = 1 IF @ (T) = 1 L=-L; M=-M; GOTO 500	
480	IF L=0 IF @ (S) = 1 L=-1; M=0; GOTO 500	
485	IF L=0 IF @ (T) = 1 L=1; M=0; GOTO 500	
490	IF M=0 IF @ (S) = 1 M=-M; L=0; GOTO 500	
495	IF M=0 IF @ (T) = 1 M=1; L=0	
500	J=J+L; K=K+M	
505	IF J<1 GOTO 550	
510	IF J>8 GOTO 550	
515	IF K<1 GOTO 550	
520	IF K>8 GOTO 550	
525	GOTO 450	
550	K=32-8xK; J=12xJ-41	
560	NT=50	
565	FOR B=1 TO 50	
570	BOX J, K, 11, 7, 3	
575	NEXT B	

Line #

Statement(s)

Comments

580 NT=3;GOTO 325

800 FOR B=11 TO 88

805 C=B÷10-1;D=RM-1

820 C=-29+C×12;D=24-D×8

830 IF@ (B)=1 B.OX C,D,7,5,
3; @ (B)=0

840 IF PX(C,D)=1 @ (1)=@ (1)+1

850 NEXT B

855 NT=50;CY=40;CX=-35

865 PRINT "F I N I S H

870 NT=3; CLEAR

875 IF @ (1)>10 GOTO 900

880 PRINT "EUREKA!

885 CY=0;PRINT "YOU HAD ",
#2,A,"ATOMS890 PRINT "YOU USED ",#1,
@ (2)-2," RAYS

895 GOTO 250

900 PRINT " SORRY

910 GOTO 250

USE OF SHADY AREA IS FOR 2ND OR
MORE LINES OF MULTI-LINE STATEMENTSDO NOT ENTER A SPACE BETWEEN LINE #
AND STATEMENT, THIS IS DONE BY THE UNIT

Page ____ of ____

Date _____

By _____

```

3  : RETURN
4  . FREQUENCIES
5  . ROBERT HOOD
6  . AUGUST 1979
8  NT = 0
10 CLEAR; PRINT "BALLY
    TONE FREQUENCIES
20 PRINT "COMPUTES FREQU
    ENCY OF TONE
30 PRINT "REGISTER A B O
    R C IF VALUE OF
40 PRINT "MASTER & TONE
    REGISTERS
50 PRINT "ARE KNOWN. ALS
    O COMPUTES
60 PRINT "SETTINGS OF TO
    NE REGISTER
70 PRINT "FOR A DESIRED
    FREQUENCY
80 PRINT "IF MASTER REGI
    STER VALUE
90 PRINT "IS KNOWN
100 PRINT "FOR FREQUENCY
    CALC INPUT 1
110 INPUT "FOR SETTING TO
    NE INPUT 2 ? "A
120 IF A = 1 GOTO 150
130 IF A = 2 GOTO 300
140 GOTO 100
150 CLEAR; INPUT "MASTER C
    OUNTER VALUE ? "M
160 INPUT "TONE COUNTER
    VALUE ? "T
170 F = 1000000 / ((M + 1) * 1.12))
    ; H = RM * 100 / ((M + 1) * 1.12))

```


172 $G_1 = 10000 \div (T+1); I = RM \times$
10000 $\div (T+1)$
174 $F = F \times G + G \times H \div 9 + F \times I \div 9 +$
 $H \times I \div 9$
190 PRINT; PRINT "FREQUENCY
Y IS", F, "HERTZ"
200 INPUT "INPUT 1 TO CON
TINUE, CALC. "Z
210 IF Z = 1 CLEAR; GOTO 100
220 STOP
300 CLEAR; INPUT "INPUT DE
SIRED FREQUENCY?" F
310 $R = 10000 \div F; V = RM \times 10 \div F$
320 $S = 89$
330 PRINT; INPUT "SET MAST
ER COUNTER VALUE?" M
340 $S = R \times S \div (M+1) + V \times S \div (M$
 $\times 9)$
350 PRINT; PRINT "FOR FREQ
UENCY OF", F
360 PRINT "MASTER COUNTER
", M
370 PRINT "TONE VALUE IS
", S
380 PRINT; PRINT "INPUT 1
TO CONTINUE
390 INPUT "INPUT 2 TO STO
P ?" J
400 IF J = 1 CLEAR; GOTO 100
410 STOP

GUESSING GAME (4)

9
?

```
1 NT=0
2 PRINT
3 PRINT
4 PRINT
5 PRINT "GUESSING GAME(4)"
6 PRINT "BY BRET DABEL AND VINCE GARZOLI"
7 PRINT
8 PRINT
9 PRINT
10 A=RND (32,000)
20 INPUT "YOUR GUESS:" B
30 IF A=B PRINT B, "IS RIGHT!"
40 IF A>B PRINT "MORE↑"
50 IF A<B PRINT "LESS↓"
60 INPUT "PLAYER 2 BGUESS:" B
70 IF A=B PRINT B, "IS RIGHT!"
80 IF A>B PRINT "MORE↑"
90 IF A<B PRINT "LESS↓"
100 INPUT "PLAYER 3 GUESS:" B
110 IF A=B PRINT B, "IS RIGHT!"
120 IF A>B PRINT "MORE↑"
130 IF A<B PRINT "LESS↓"
140 INPUT "PLAYER 4 GUESS:" B
150 IF A=B PRINT B, "IS RIGHT!"
160 IF A>B PRINT "MORE↑"
170 IF A<B PRINT "LESS↓"
180 IF A=B GOTO 10
190 GOTO 20
```

AUGUST 2, 1979

DEAR BOB,

If Rory Wohl's computer works with the changes in the BANGMAN program that he suggested on page 58 of the ARCADIAN it has a faulty logic chip. 9

He suggests the line should read:

2000 E=E+1; IF E=9 GOTO 9000; IF Q # 1 GOSUB 9600+(E x 10)

T Q is a flag that is set to either 0 or 1. If it is set to 0 it sends the program to the man drawing routine at 9600, 9610, 9620, etc.

E is a counter that is to be incremented ONLY if the guess is wrong. It is NOT to be incremented if the guess is correct or if the letter has been previously used. So the portion of the program line, E=E+1, must follow the IF Q # 1 GOSUB 9600+(E x 10) statement.

Now, if the 'IF' portion of the IF E=9 GOTO 9000 statement is not satisfied the program defaults to the next numbered program line. The way Rory proposes, E would be incremented each time a guess is made, right or wrong. The program WOULD never reach the man drawing routine statement because it can't go past the first IF statement until E=9 at which time the program goes to 9000, draws the gun and shoots the man that never gets drawn on the screen.

So line 2000 MUST remain exactly as was originally written or the program will not work as intended:

↓ 2000 IF Q # 1 GOSUB 9600+(E x 10); E=E+1; IF E=9 GOTO 9000

NOW FOR SOME OTHER STUFF. I'M WRITING THIS ON MY TRS-80 WHICH DOESN'T HAVE UPPER AND LOWER CASE. HOWEVER, I FOUND THAT IF I HOLD THE SHIFT KEY DOWN WHEN I'M TYPING THAT THE LETTERS TRANSMITTED TO MY TRENDATA SELECTRIC COME OUT IN LOWER CASE. IT MAKES THE TYPING QUITE SLOW AND I'M NEVER SURE I HAVE A LOWER CASE LETTER UNTIL I SEE IT PRINTED BECAUSE THE SCREEN DISPLAYS ONLY UPPER CASE.

BY THE WAY, I ORDERED THE SERVICE MANUAL AND THE HACKER'S MANUAL SOME TIME AGO. YOU SENT THE SERVICE MANUAL RIGHT AWAY BUT AS FAR AS I KNOW I HAVEN'T RECEIVED THE HACKER'S MANUAL. ONE OF THE FELLOWS HERE AT WESTERN WASHINGTON UNIVERSITY SAID HE WOULD TRY TO INTERFACE MY SELECTRIC AS SOON AS I GOT SOME SCHEMATICS

ISSUE NUMBER 8 IS GREAT. THEY ARE GETTING BETTER ALL THE TIME.

CHUCK THOMKA'S WRITE-UP ON THE MUSIC SYNTHESIZER WAS EXCEPTIONAL. I DON'T KNOW HOW SOME PEOPLE CAN COME UP WITH SO MUCH INFORMATION. THE TOUCH TONE SIMULATOR WORKS PERFECTLY AND THE SOUND GRAPH IS A CLEVER WAY OF DISPLAYING TONE TESTING DATA. I MADE A PROGRAM SOMETHING LIKE THAT WHEN I FIRST TRIED OUT THE &(16), ETC. REGISTERS BUT I DIDN'T PUT IN THE GRAPH.

IN RESPONSE TO YOUR QUESTIONS:

1. YES.
2. PROBABLY NOT.
3. I'M NOT SURE. IT DEPENDS ON THE LANGUAGE CAPABILITIES AND THE PERIPHERAL POSSIBILITIES. I REALLY WOULD PREFER ZGRASS BASIC, THAT'S ONE OF THE REASONS I'VE BEEN WAITING, AND I'M GOING TO WANT A DISC DRIVE OR TWO AND AN RS-232 INTERFACE FOR MY PRINTER. ALSO, I WOULD PREFER TO HAVE SOME STRONG SUPPORT FROM A COMPANY THAT'S GOING TO BE AROUND FOR A WHILE BUT SOMETIMES ONE HAS TO TAKE WHAT IS AVAILABLE NOT WHAT HE WOULD LIKE TO HAVE.
4. MY NEW BALLY ARCADE SERIAL NUMBER IS: #20359, MODEL BPA 1100.

SINCERELY,



ERNEST W. SAMS
248 SOUTH FOREST STREET
BELLINGHAM, WA 98225

July 12, 1979

HI, BOB

I RECENTLY RECEIVED MY FIRST COPIES OF ARCADIAN,
AND FOUND IT MOST INFORMATIVE, KEEP UP THE
GOOD WORK.

I THOUGHT I MIGHT HELP ALL THE "ARCADIAN POKERS",
WITH A PROGRAM THAT LOADS MACHINE'S INSTRUCTIONS
INTO THE @ STRING, IN THIS WAY YOU CAN CALL
SEVERAL MACHINE LANGUAGE SUBROUTINES FROM
YOUR BASIC MAINLINE, WITHOUT HAVING TO MAKE
THEM BASIC STATEMENTS. BELOW IS THE PROGRAM
LISTING, AND SAMPLE DATA.

```

5 NT=0
10 A=20180; B=A; FOR K=0 TO 13
20 INPUT @ (K); NEXT K
30 FOR K=0 TO 13; CLEAR
40 CY=0; PRINT K, @ (K)
50 D=KP; IF D=31 GOTO B0
60 IF D=57 GOTO 90
70 GOTO 50
80 INPUT "CHANGE="; L; @ (K)=L
90 NEXT K
100 A=B; FOR K=0 TO 13
110 % (A)=@ (K); A=A+Z; NEXT K
120 IF & (Z)=8 GOTO 30
130 C=20180; GOSUB 160
140 C=20190; GOSUB 160
150 GOTO 120
160 CLEAR; CALL (C); RETURN

```

	@ STRING	SAMPLE DATA MACHINE LEVEL CODE TO BE INPUT	TO BE POKE INTO MEMORY ADDRESS
SUBROUTINE 1	0 =	-43 05 FF	20180
	1 =	12341 35 30	20182
	2 =	19480 18 4C	20184
	3 =	31640 31 59	20186
	4 =	13871 31 09	20188
SUBROUTINE 2	5 =	43 05 FF	20190
	6 =	53 35 00	20192
	7 =	27672 18 6C	20194
	8 =	20200 E8 4E	20196
	9 =	13871 31 09	20198
	10 =	21057 41 52	20200
	11 =	16707 43 41	20202
	12 =	18756 14 49	20204
	13 =	20033 41 4E	20206

STATEMENT COMMENTS

10-20 INPUT MACHINE INSTRUCTIONS
 INTO @ STRING.

30-90 — EDIT ROUTINE —
 HIT "STEP" KEY (D=57) TO STEP
 THROUGH MACHINE INSTRUCTIONS.
 HIT "ERASE" KEY (D=31) TO CHANGE
 MACHINE INSTRUCTION.

100-110 POKE @ STRING INTO MEMORY.

120 HIT "GOTO" KEY TO BRANCH
 BACK TO EDIT ROUTINE AT ANY TIME.

130-140 INITIALIZE STARTING ADDRESSES
 FOR SUBROUTINE #2.

160 CALL SUBROUTINE #2

Howdy

MODIFICATION TO BIG LETTER PRINT - P. 45; ARCADIAN

65° A = 20190 . START OF DISPLAY AREA
70 K = KP
80 IF K = 13 GOTO 100
90 %A) = K; A = A + 1; GOTO 70
100 %A) = 0 . DISPLAY SHUTS OFF IF 0 ENCOUNTERED
105 CLEAR
110 CALL (B); GOTO 65
120 %A) = X; A = A + 2; RETURN . SAME AS ORIGINAL

the ASCII values of K get poked into
the display area 8 BITS at a time.
the subroutine shuts off if it sees a 0.

I sent you some info on %A) into screen
memory that was in error. I was
poking into 1500, and it appeared on
the screen ~~by some~~ for some reason.
I've revised ~~it~~ and included again.

Renzi

ANY EXPANSION CAPABILITIES WOULD BE WELCOME!

MODIFIED
ARCADIAN
PAGE 45

* @ & # \$? ! !

1 2 3 4 5 6 7 8 9 0

1 2 3 4 5 6 7 8 9 0

MORE INFORMATION ON THE SOUND SYNTHESIZER

By: Chuck Thomka

Whenever "RESET" is pushed, the &16 to &23 registers are set to fixed values. This also happens at POWER TURN ON. &16 is set to 71 and &17 through &23 are set to zero. [This ~~group of registers~~ is called the DEFAULT VALUE]

Since pushing most keys on the keypad will generate a sound, one of the voices must be used. This means that since most keys have unique tones when pushed, they must be loading unique values into one or more of the registers. The voice used is the 'A' counter. Each key, when pushed, puts a value into the &17 register that the 'A' counter will count up to. It also will put value 15 into the &22 register, that will adjust the 'A' volume to maximum so that the resultant frequency can be heard. At the end of the time of outputting the tone the &17 and &22 are both put back to zero.

At anytime that the computer is stopped, the &16 register will be set to 71, and &17 and &22 will be set to zero. This may affect some results of sound effects in programs where you want those registers to be left at some other values. All the other registers will be as they were last adjusted to, so remember this if you still have a tone or noise remaining on after the computer has stopped.

Later in this article is a table of all the sound generating keys, their &17 values, the resultant frequencies, and any special notes about them.

The keys that do not generate sounds are \div , X, + and -. These keys will modify the sounds created by the other keys if the modifying keys are used just prior to the normal sound keys.

The \div (divide) key will make the sound one octave lower in frequency than normal. This is done by temporarily making the master counter &16 count twice as far. So while &16 is normally at 71, for this one note, it will be set to 143. As soon as the note has finished &16 will again return to 71 unless the next note is also preceded by a \div .

The X (multiply) key will make the sound one octave higher in frequency. This is done by making &16 equal 35 for the time the concerned note is sounding. At the end of which, &16 will again return to 71.

The + (plus) and - (minus) keys are only used in conjunction with the numbers 1 through 7. This was arranged so that the plus and minus sign would be meaningful in playing musical sharps or flats in the Bally mentioned 3 octave musical scale.

Another thing to mention is the "Note Timer" or NT; for each number of NT, the notes played will be approximately 17 milliseconds long. An NT=0 results in no sound, while the maximum number of NT=255 results in about a 4.335 second (.017X255=4.335) note. It is interesting to observe the effect of printing a character on the TV that is followed by a 0. The "player piano" program in the Bally Programmed Instruction Course uses the 0 (zero) to extend the duration of a played note. How it does this, (for only the one note to be played) it takes the note timer and increases it an additional NT quantity for each 0 following the note to be

heard. For example, say we are to play a note while the NT=10, and that this note is followed by 3 zeros, the resultant NT will be 40. After playing that modified NT, the NT will again return to normal (10, in this example) until called upon again.

A funny thing about this method of extending the duration of a played note is that you still cannot play any note longer than 4.335 seconds. This is because if you had a note timer extended by way of using zeros after a printed character, and it would result in an NT>255, the final result would probably be less than 255. To explain what I mean, you have to know about binary numbers and that the NT register is only 8 bits wide. If, for example, we had an NT of 50 and that some program that we are running is to print a character followed by 5 zeros, we would expect a temporary NT result of 300 ($1+5=6, 6 \times 50=300$) but an 8 bit register's maximum bit count is only 255, while a binary conversion of decimal 300 requires 9 bits (1 0010 1100). The result is that only the least 8 bits (0010 1100) will be loaded into the NT register, so NT will temporarily be 44. This, you see is a lot shorter than we had at first expected and even shorter than the normal NT of 50.

Finishing up, I am now giving you a table of all the possible frequencies generated by printing to the TV, the characters that will create them and also resultant frequencies if the character is preceeded by a \div or X sign.

RESULTANT FREQUENCY

$R(17)$ VALUE	CHARACTER(S)	NORMAL $R(16)=71$	\div PREFIX $R(16)=143$	\times PREFIX $R(16)=35$
225	!	54.63 Hz	27.32 Hz	109.26 Hz
212	"	57.97	28.98	115.93
200	#	61.43	30.71	122.85
189	\$	64.98	32.49	129.97
178	%	68.98	34.49	137.95
168	&	73.06	36.53	146.12
159	' (APOST.)	77.17	38.58	154.34
150	(81.77	40.88	163.53
141)	86.95	43.47	173.90
133	*	92.14	46.07	184.28
119	, (COMMA)	102.89	51.45	205.78
106	. (PERIOD)	115.39	57.70	230.78
100	/ Y -1	122.25	61.12	244.49
94	Z 1	129.97	64.98	259.93
89	[+1 -2	137.19	68.59	274.37
84	\ 2	145.26	72.63	290.51
79] +2 -3	154.34	77.17	308.67
74	↑ 3	164.62	82.31	329.25
70	← 4	173.90	86.95	347.80
66	↓ +4	184.28	92.14	368.56
62	→ 5	195.98	97.99	391.96
59	+5 -6	205.78	102.89	411.56
55	6	220.48	110.24	440.96
52	+6 -7	232.96	116.48	465.92
49	7	246.94	123.47	493.87
46	+7 8	262.70	131.35	525.40

RESULTANT FREQUENCY

$\mathcal{E}(17)$ VALUE	CHARACTER(S)	NORMAL	\div PREFIX	\times PREFIX
		$\mathcal{E}(16) = 71$	$\mathcal{E}(16) = 143$	$\mathcal{E}(16) = 35$
44	9	274.37 HZ	137.19 HZ	548.75 HZ
41	:	293.97	146.99	587.94
39	;	308.67	154.34	617.34
37	<	324.92	162.46	649.83
34	=	352.77	176.38	705.53
32	>	374.15	187.07	748.29
31	?	385.84	192.92	771.68
29	@	411.56	205.78	823.12
27	A	440.96	220.48	881.91
26	B	457.29	228.64	914.58
24	C	493.87	246.94	987.74
23	D	514.45	257.23	1028.90
21	E	561.22	280.61	1122.44
20	F	587.94	293.97	1175.89
19	G	617.34	308.67	1234.68
18	H	649.83	324.92	1299.66
17	I	685.93	342.97	1371.87
16	J	726.28	363.14	1452.57
15	K	771.68	385.84	1543.35
14	L	823.12	411.56	1646.24
13	M	881.91	440.96	1763.83
11	N	1028.90	514.45	2057.80
10	O	1122.44	561.22	2244.87
9	P	1234.68	617.34	2469.36
8	Q	1371.87	685.93	2743.73
7	R	1543.35	771.68	3086.70

RESULTANT FREQUENCY

&(17) VALUE	CHARACTER(S)	NORMAL	÷ PREFIX	x PREFIX
		&(16) = 71	&(16) = 143	&(16) = 35
6	S	1763.83 HZ.	881.91 HZ.	3527.66 HZ.
5	T	2057.80	1028.90	4115.60
4	U	2469.36	1234.68	4938.72
3	V	3086.70	1543.35	6173.40
2	W	4115.60	2057.80	8231.20
1	X	6173.40	3086.70	12346.81